pandas-utility-sneak-peek / functional_testing_demo_notebooks
/ func_test_summary_dfsummarize_5_PPSF_Data.ipynb

techds functional_testing > new and updated Jupyter notebooks bc01803 · 3 weeks ago

2419 lines (2419 loc) · 86.8 KB

Preview Code Blame Raw

In [ ]:
```
'''

Let's explore the functionality of the dfsummarize method in the pdutils package for a small

dataset, namely, the PPSF AllHomes Data. We will be using a Jupyter notebook in VS Code. Please

note that although this notebook does not represent an exhaustive examination of the dfsummarize

module's functionality, it does highlight some key features of the method. Additionally, you may

find it useful to review another use case for dfsummarize--see the Jupyter notebook and video

comprising the application of pdutils-dfsummarize to US Consumer Complaints Data.

'''
```

In [2]:
```
'''

Let's start with importing all the necessary modules in the first cell. Once the modules are

imported, the cell output confirms with a message that all imports have been imported!

'''

import numpy as np
import pandas as pd
import time
from pdutils.summary.dfsummarize import dfsummarize
from pdutils.utilities.time_func import time_func

print('All imports have been imported!')
```

```
All imports have been imported!
```

In [3]:
```
'''

In this cell, we use the variable `f` to point to the PPSF AllHomes data which

has over a million records and takes a little over 50 MB of storage space.

'''

f = '../../../../../Documents/datasets/data_RE_v3_ppsqft_allhomes.csv'
```

In [4]:
```
'''

In the next cell, you will read the given data as a Pandas dataframe (df). Use

dtype=np.object_ in read_csv() if you need to designate `object` data type for columns. This

is helpful, especially, when your dataframe columns contain mixed data types.

'''

dfx = pd.read_csv(f)
print('Specified dataset read. Shape and column list is as follows:')
print(dfx.shape)
print(dfx.columns)
dfx.head()
```

```
Specified dataset read. Shape and column list is as follows:
(1255915, 6)
Index(['date', 'reg', 'zip', 'city', 'state', 'ppsf'], dtype='object')
```

Out[4]:

	date	reg	zip	city	state	ppsf
0	2010-02-01	612	612	Thoreau	NM	NaN
1	2010-03-01	612	612	Thoreau	NM	NaN
2	2010-04-01	612	612	Thoreau	NM	NaN
3	2010-05-01	612	612	Thoreau	NM	NaN
4	2010-06-01	612	612	Thoreau	NM	NaN

In [5]:
```
'''

Next, set the option to display all columns.

Also, set the option to display all rows.

'''

pd.options.display.max_columns = None
pd.options.display.max_rows = None
```

In [6]:
```
'''

The dfsummarize function may be run in any one of two modes: simple and comprehensive.

The simple mode in dfsummarize provides a quick and easy way to summarize the given

dataframe. The comprehensive mode comprises additional functionality, including dataframe

column search, VIFs, frequency table, and more. Numbers are displayed in scientific

notation by default.

'''

print('Summarizing dataframe with the simple mode (simple=True)...')
summary = time_func(dfx.dfsummarize, simple=True, sort_mtable=False)
mtab, _ = summary
mtab
```

```
Summarizing dataframe with the simple mode (simple=True)...
    Dataframe shape (r, c): (1255915, 6)
    Features (cols) list  : ['date', 'reg', 'zip', 'city', 'state', 'ppsf']
--------------------------------

Wall time elapsed: 0.1716 seconds
Wall time elapsed: 0.0 minutes
CPU time elapsed : 0.1564 seconds
CPU time elapsed : 0.0 minutes
```

Out[6]:

	feature	dtype	count	mean	std	min	25%	50%	75%	max
0	date	object	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	reg	int64	1.26e+06	4.91e+04	3.00e+04	6.12e+02	2.41e+04	4.47e+04	7.73e+04	9.99e+04
2	zip	int64	1.26e+06	4.91e+04	3.00e+04	6.12e+02	2.41e+04	4.47e+04	7.73e+04	9.99e+04
3	city	object	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	state	object	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	ppsf	float64	1.04e+06	1.64e+02	1.34e+02	1.05e+01	9.43e+01	1.27e+02	1.83e+02	2.80e+03

In [10]:
```
'''

Let's run the simple summary again, with numeric data being displayed in standard notation

as integers. Pass count_null=True to calculate and display total missing value count.

'''

print('Summarizing dataframe with the simple mode (simple=True) in standard notation...')
summary = time_func(dfx.dfsummarize, simple=True, sort_mtable=False, to_sci_no=False,
                    rounding=0, count_null=True)
mtab, _ = summary
mtab
```

```
Summarizing dataframe with the simple mode (simple=True) in standard notation...
    Dataframe shape (r, c): (1255915, 6)
    Features (cols) list  : ['date', 'reg', 'zip', 'city', 'state', 'ppsf']
    Missing value count   : 216973
--------------------------------

Wall time elapsed: 0.3224 seconds
Wall time elapsed: 0.01 minutes
CPU time elapsed : 0.3073 seconds
CPU time elapsed : 0.01 minutes
```

Out[10]:

	feature	dtype	count	mean	std	min	25%	50%	75%	max
0	date	object	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	reg	int64	1255915	49104	29985	612	24148	44667	77320	99901
2	zip	int64	1255915	49104	29985	612	24148	44667	77320	99901
3	city	object	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	state	object	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	ppsf	float64	1038942	164	134	10	94	127	183	2799

In [11]:
```
'''

Run the dfsummarize function, using all the default parameters for arguments except for some.
We will use the Pandasmic variant of the method, where we append dfsummarize as a method to the
DataFrame class. For faster processing, pass date_col_ls=[], num_col_ls=[], check_bool=False,
and stats=False in the dfsummarize method. You can measure wall and CPU times elapsed by wrapping
the function in time_func().

The main summary table (mtab) in the cell output here displays 8 columns.

These columns from left to right are, Feature (or column name), C-Type (or column data type),
V-type (or column value data type), MLS (or maximum length of column values represented as
strings), the number of non-null values, the number of unique values, the number of null values,
and the search column index list.

The first row (with row index=0), in the main table, for example, contains the `city` feature
with object data type associated with the column and string data type representing the column
values. This row also shows that the city column contains 25 characters in its longest string
value, approximately 1.26 million non-null entries, 5340 unique values, and zero null values.

'''
summary = time_func(dfx.dfsummarize, date_col_ls=[], num_col_ls=[], check_bool=False, stats=False)
mtab, _ = summary
mtab
```

```
   Dataframe shape (r, c): (1255915, 6)
   Features (cols) list  : ['date', 'reg', 'zip', 'city', 'state', 'ppsf']
   Missing value count   : 216973

*** Searching data for missing values...
*** Eliminating missing values...
*** Row count has dropped from 1255915 to 1038942 due to null removal.
*** Post-row-removal null count across columns is 0.

Legend   :
   C-Type: Column data type
   V-Type: Column value data type
   MLS   : Maximum length of [column values converted into] string data type
   NIL   : Null index list
   SCIL  : Search column index list (list of col-specific row indexes that match the search criterion)

Wall time elapsed: 7.9692 seconds
Wall time elapsed: 0.13 minutes
CPU time elapsed : 7.418 seconds
CPU time elapsed : 0.12 minutes
```

Out[11]:

	Feature	C-Type	V-Type	MLS	#NonNull	#Unique	#Null	SCIL
0	city	object	[str]	2.50e+01	1.26e+06	5.34e+03	0.00e+00	N/A
1	date	object	[str]	1.00e+01	1.26e+06	1.15e+02	0.00e+00	N/A
2	ppsf	float64	[float64]	7.00e+00	1.04e+06	6.43e+04	2.17e+05	N/A
3	reg	int64	[int64]	5.00e+00	1.26e+06	1.09e+04	0.00e+00	N/A
4	state	object	[str]	2.00e+00	1.26e+06	5.10e+01	0.00e+00	N/A
5	zip	int64	[int64]	5.00e+00	1.26e+06	1.09e+04	0.00e+00	N/A

In [12]:
```
'''

If you want to see Min, Median, and Max values for all the numeric columns in your data, but

don't want the algorithm to compute potential outlier values (or POVs), and variance inflation
```

```
factors (or VIFs) in the Main table, remove the stats=False argument from the dfsummarize
method. Also, add show_POV=False and show_VIF=False arguments in the dfsummarize method. To
avoid computing outlier values, you can also pass an empty list to the pov_col_ls argument
in dfsummarize. Here, we take the first approach. Allow the features to be displayed in the
order in which they appear in the data by passing sort_mtable=False.
Note that the values in region and zip columns, which appear to contain the same data, are
expressed as integer data type. Hence, these columns show the corresponding minimum, median,
and maximum values as part of the stats array.
'''

summary = time_func(dfx.dfsummarize, date_col_ls=[], num_col_ls=[], sort_mtable=False,
                    show_POV=False, show_VIF=False)
mtab, _ = summary
mtab
```

```
   Dataframe shape (r, c): (1255915, 6)
   Features (cols) list  : ['date', 'reg', 'zip', 'city', 'state', 'ppsf']
   Missing value count   : 216973

*** Searching data for missing values...
*** Eliminating missing values...
*** Row count has dropped from 1255915 to 1038942 due to null removal.
*** Post-row-removal null count across columns is 0.
*** Getting ready to compute the stats array...
*** Using ['reg', 'zip', 'ppsf'] as the list of numerical columns for the stats array.

Legend   :
   C-Type: Column data type
   V-Type: Column value data type
   MLS   : Maximum length of [column values converted into] string data type
   NIL   : Null index list
   SCIL  : Search column index list (list of col-specific row indexes that match the search criterion)
   Mid   : Median
   POV   : Potential outlier value(s) for z-score thresh=1.5
   VIF   : Variance inflation factor(s)

Wall time elapsed: 7.5023 seconds
Wall time elapsed: 0.13 minutes
CPU time elapsed : 7.4279 seconds
CPU time elapsed : 0.12 minutes
```

Out[12]:

	Feature	C-Type	V-Type	MLS	#NonNull	#Unique	#Null	SCIL	Min	Mid	Max	POV	VIF
0	date	object	[str]	1.00e+01	1.26e+06	1.15e+02	0.00e+00	N/A	NaN	NaN	NaN	NaN	NaN
1	reg	int64	[int64]	5.00e+00	1.26e+06	1.09e+04	0.00e+00	N/A	6.12e+02	4.41e+04	9.99e+04	N/A	N/A
2	zip	int64	[int64]	5.00e+00	1.26e+06	1.09e+04	0.00e+00	N/A	6.12e+02	4.41e+04	9.99e+04	N/A	N/A
3	city	object	[str]	2.50e+01	1.26e+06	5.34e+03	0.00e+00	N/A	NaN	NaN	NaN	NaN	NaN
4	state	object	[str]	2.00e+00	1.26e+06	5.10e+01	0.00e+00	N/A	NaN	NaN	NaN	NaN	NaN
5	ppsf	float64	[float64]	7.00e+00	1.04e+06	6.43e+04	2.17e+05	N/A	1.05e+01	1.27e+02	2.80e+03	N/A	N/A

In [14]:

```
'''
To see the complete stats array, remove the show_POV=False and show_VIF=False arguments from the
dfsummarize method. Pass to_sci_no=False to display numbers (converted to scientific notation by
default) in standard notation.
'''

summary = time_func(dfx.dfsummarize, date_col_ls=[], num_col_ls=[], to_sci_no=False,
                    sort_mtable=False)
mtab, _ = summary
mtab
```

```
   Dataframe shape (r, c): (1255915, 6)
```

```
    Features (cols) list  : ['date', 'reg', 'zip', 'city', 'state', 'ppsf']
    Missing value count   : 216973

*** Searching data for missing values...
*** Eliminating missing values...
*** Row count has dropped from 1255915 to 1038942 due to null removal.
*** Post-row-removal null count across columns is 0.
*** Getting ready to compute the stats array...
*** Using ['reg', 'zip', 'ppsf'] as the list of numerical columns for the stats array.
*** Parsing numerical data ['reg', 'zip', 'ppsf'] for potential outlier values (POVs)...
**************************************************
>>> Time elapsed : 0.0 sec
>>> Drop values list generated...
>>> Time elapsed : 0.6067 sec
>>> Drop values list generated...
>>> Time elapsed : 1.1346 sec
>>> Drop values list generated...
>>> Time elapsed : 1.6037 sec
>>> Drop values list generated...
>>> Time elapsed : 2.0121 sec
>>> Drop values list generated...
>>> Time elapsed : 2.3441 sec
>>> Drop values list generated...
>>> Time elapsed : 2.6115 sec
>>> Drop values list generated...
>>> Time elapsed : 2.8715 sec
>>> Drop values list generated...
>>> Time elapsed : 3.0696 sec
>>> Drop values list generated...
>>> Time elapsed : 3.2219 sec
>>> Drop values list generated...
>>> Time elapsed : 3.3503 sec
>>> Drop values list generated...
>>> Time elapsed : 3.4569 sec
>>> Drop values list generated...
>>> Time elapsed : 3.5484 sec
>>> Drop values list generated...
>>> Time elapsed : 3.6266 sec
>>> Drop values list generated...
>>> Time elapsed : 3.6951 sec
>>> Drop values list generated...
>>> Time elapsed : 3.7556 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8068 sec
>>> Drop values list generated...
>>> Time elapsed : 3.85 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8875 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9204 sec
>>> Drop values list generated...
>>> Time elapsed : 3.948 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9711 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9914 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0096 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0249 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0388 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0509 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0621 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0713 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0792 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0859 sec
>>> Drop values list generated...
>>> Time elapsed : 4.092 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0981 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1044 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1103 sec
>>> Drop values list generated
```

```
>>> Drop values list generated...
>>> Time elapsed : 4.1163 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1224 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1287 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1346 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1403 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1459 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1513 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1563 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1607 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1648 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1681 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1711 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1738 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1762 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1785 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1814 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1836 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1855 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1871 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1886 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1898 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1909 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1918 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1925 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1932 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1938 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1943 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1948 sec
Breaking as no index was found at iter 63 ...
**********************************************************
>>> Time elapsed : 0.0 sec
>>> Drop values list generated...
>>> Time elapsed : 0.5425 sec
>>> Drop values list generated...
>>> Time elapsed : 1.068 sec
>>> Drop values list generated...
>>> Time elapsed : 1.5599 sec
>>> Drop values list generated...
>>> Time elapsed : 1.9816 sec
>>> Drop values list generated...
>>> Time elapsed : 2.3138 sec
>>> Drop values list generated...
>>> Time elapsed : 2.5851 sec
>>> Drop values list generated...
>>> Time elapsed : 2.8207 sec
>>> Drop values list generated...
>>> Time elapsed : 3.014 sec
>>> Drop values list generated...
>>> Time elapsed : 3.169 sec
>>> Drop values list generated...
>>> Time elapsed : 3.2941 sec
>>> Drop values list generated...
>>> Time elapsed : 3.4007 sec
```

```
>>> Drop values list generated...
>>> Time elapsed : 3.4927 sec
>>> Drop values list generated...
>>> Time elapsed : 3.5711 sec
>>> Drop values list generated...
>>> Time elapsed : 3.6409 sec
>>> Drop values list generated...
>>> Time elapsed : 3.701 sec
>>> Drop values list generated...
>>> Time elapsed : 3.7525 sec
>>> Drop values list generated...
>>> Time elapsed : 3.7957 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8335 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8666 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8942 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9172 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9369 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9546 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9695 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9829 sec
>>> Drop values list generated...
>>> Time elapsed : 3.995 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0056 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0147 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0225 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0289 sec
>>> Drop values list generated...
>>> Time elapsed : 4.035 sec
>>> Drop values list generated...
>>> Time elapsed : 4.041 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0471 sec
>>> Drop values list generated...
>>> Time elapsed : 4.053 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0589 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0648 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0705 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0771 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0828 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0884 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0938 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0988 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1033 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1074 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1108 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1138 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1165 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1189 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1212 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1234 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1253 sec
>>> Drop values list generated
```

```
>>> Drop values list generated...
>>> Time elapsed : 4.127 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1286 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1301 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1313 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1324 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1333 sec
>>> Drop values list generated...
>>> Time elapsed : 4.134 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1347 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1353 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1358 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1362 sec
Breaking as no index was found at iter 63 ...
**************************************************
>>> Time elapsed : 0.0 sec
>>> Drop values list generated...
>>> Time elapsed : 0.5659 sec
>>> Drop values list generated...
>>> Time elapsed : 1.1377 sec
>>> Drop values list generated...
>>> Time elapsed : 1.6263 sec
>>> Drop values list generated...
>>> Time elapsed : 2.051 sec
>>> Drop values list generated...
>>> Time elapsed : 2.4199 sec
>>> Drop values list generated...
>>> Time elapsed : 2.7367 sec
>>> Drop values list generated...
>>> Time elapsed : 3.0157 sec
>>> Drop values list generated...
>>> Time elapsed : 3.2567 sec
>>> Drop values list generated...
>>> Time elapsed : 3.4679 sec
>>> Drop values list generated...
>>> Time elapsed : 3.6488 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8002 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9327 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0479 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1482 sec
>>> Drop values list generated...
>>> Time elapsed : 4.2365 sec
>>> Drop values list generated...
>>> Time elapsed : 4.31 sec
>>> Drop values list generated...
>>> Time elapsed : 4.3749 sec
>>> Drop values list generated...
>>> Time elapsed : 4.4316 sec
>>> Drop values list generated...
>>> Time elapsed : 4.4793 sec
>>> Drop values list generated...
>>> Time elapsed : 4.5206 sec
>>> Drop values list generated...
>>> Time elapsed : 4.5561 sec
>>> Drop values list generated...
>>> Time elapsed : 4.5873 sec
>>> Drop values list generated...
>>> Time elapsed : 4.6157 sec
>>> Drop values list generated...
>>> Time elapsed : 4.6403 sec
>>> Drop values list generated...
>>> Time elapsed : 4.6609 sec
>>> Drop values list generated...
>>> Time elapsed : 4.679 sec
>>> Drop values list generated...
>>> Time elapsed : 4.6946 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7082 sec
```

```
>>> Drop values list generated...
>>> Time elapsed : 4.7197 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7295 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7381 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7456 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7521 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7578 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7628 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7671 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7709 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7742 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7769 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7793 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7815 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7833 sec
>>> Drop values list generated...
>>> Time elapsed : 4.785 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7865 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7878 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7889 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7899 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7908 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7916 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7923 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7929 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7934 sec
Breaking as no index was found at iter 53 ...
*** There are no missing elements in the target list...returning an empty index list.
*** Done POVs!
*** Attempting to compute variance inflation factors (VIFs)...
/Users/ccss/opt/anaconda3/envs/pdu310/lib/python3.10/site-packages/statsmodels/stats/outliers_influence.py:
198: RuntimeWarning: divide by zero encountered in scalar divide
  vif = 1. / (1. - r_squared_i)
Legend   :
   C-Type: Column data type
   V-Type: Column value data type
   MLS   : Maximum length of [column values converted into] string data type
   NIL   : Null index list
   SCIL  : Search column index list (list of col-specific row indexes that match the search criterion)
   Mid   : Median
   POV   : Potential outlier value(s) for z-score thresh=1.5
   VIF   : Variance inflation factor(s)

Wall time elapsed: 25.0285 seconds
Wall time elapsed: 0.42 minutes
CPU time elapsed : 29.4548 seconds
CPU time elapsed : 0.49 minutes
```

Out[14]:

	Feature	C-Type	V-Type	MLS	#NonNull	#Unique	#Null	SCIL	Min	Mid	Max	POV	VIF
0	date	object	[str]	10	1255915	115	0	N/A	NaN	NaN	NaN	NaN	NaN
1	reg	int64	[int64]	5	1255915	10921	0	N/A	612.00	44095.00	99901.00	[612, 623, 692, 693, 725, 745, 778, 791, 802, ...	inf

[612, 623

2	zip	int64	[int64]	5	1255915	10921	0	N/A	612.00	44095.00	99901.00	[612, 623, 692, 693, 725, 745, 778, 791, 802, ...	inf
3	city	object	[str]	25	1255915	5344	0	N/A	NaN	NaN	NaN	NaN	NaN
4	state	object	[str]	2	1255915	51	0	N/A	NaN	NaN	NaN	NaN	NaN
5	ppsf	float64	[float64]	7	1038942	64309	216973	N/A	10.46	126.92	2799.42	[10.46, 10.58, 11.08, 11.59, 11.73, 11.98, 11....	1.882

In [15]:
```
'''
It appears that `reg` and `zip` represent the same geographical location. You can use the
pdutils-dforder method to select and order specific columns in the dataframe.
'''

from pdutils.tidying.dforder import dforder
dfz = dfx.dforder(columns=['date', 'city', 'state', 'zip', 'ppsf'])
print(dfz.shape)
print(dfz.columns)
dfz.head()
```

```
(1255915, 5)
Index(['date', 'city', 'state', 'zip', 'ppsf'], dtype='object')
```

Out[15]:

	date	city	state	zip	ppsf
0	2010-02-01	Thoreau	NM	612	NaN
1	2010-03-01	Thoreau	NM	612	NaN
2	2010-04-01	Thoreau	NM	612	NaN
3	2010-05-01	Thoreau	NM	612	NaN
4	2010-06-01	Thoreau	NM	612	NaN

In [17]:
```
'''
Note from the output earlier, the ZIP code column contains anomalous values such as 612.
To identify the row index or indexes where such anomalies occur, you can use the search
feature (search_col and search_str arguments) in dfsummarize while summarizing the subset
(dfz) created with dforder() previously. The list of row indexes associated with the
given search_str (in this case, it is a numeric value) can be seen under the SCIL column
in the main summary table. The search returns an empty list if no row indexes are found
containing the given search_str. Once you are ready to clean the data, you can apply
the pdutils-dftidy method to replace the anomalous values in your dataframe. Use the
scientific notation (default) to display numeric data in the summary.
'''

summary = time_func(dfz.dfsummarize, sort_mtable=False,
                    search_col='zip', search_str=612)
mtab, _ = summary
mtab
```

```
   Dataframe shape (r, c): (1255915, 5)
   Features (cols) list  : ['date', 'city', 'state', 'zip', 'ppsf']
   Missing value count   : 216973
   Search col | op | str : zip | == | 612
   Row idxs in search col: ['N/A', 'N/A', 'N/A', [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1
6, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42,
43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
```

```
70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96,
97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114], 'N/A']

*** Searching data for missing values...
*** Eliminating missing values...
*** Row count has dropped from 1255915 to 1038942 due to null removal.
*** Post-row-removal null count across columns is 0.
*** Getting ready to check for datetime pattern in ['date', 'city', 'state', 'zip', 'ppsf'].
>>> Checking for datetime pattern in 'date' using %Y-%m-%d as the date format...
    Done in 23.34 sec.
>>> Checking for datetime pattern in 'city' using %Y-%m-%d as the date format...
    Done in 20.43 sec.
>>> Checking for datetime pattern in 'state' using %Y-%m-%d as the date format...
    Done in 20.59 sec.
>>> Checking for datetime pattern in 'zip' using %Y-%m-%d as the date format...
    Done in 21.67 sec.
>>> Checking for datetime pattern in 'ppsf' using %Y-%m-%d as the date format...
    Done in 21.7 sec.
*** Column that comprises data with a likely datetime pattern: ['date'].
*** Done converting column 'date' to datetime dtype.
*** Parsing and converting numeric columns...
*** Columns that comprise data with a likely numerical pattern: ['city', 'state', 'zip', 'ppsf'].
An exception of type ValueError occurred (Unable to parse string "Thoreau" at position 0)...Skipping conver
sion of column 'city' to numeric dtype...
An exception of type ValueError occurred (Unable to parse string "NM" at position 0)...Skipping conversion
of column 'state' to numeric dtype...
*** Done converting column 'zip' to numeric dtype.
*** Done converting column 'ppsf' to numeric dtype.
*** Getting ready to compute the stats array...
*** Using ['zip', 'ppsf'] as the list of numerical columns for the stats array.
*** Parsing numerical data ['zip', 'ppsf'] for potential outlier values (POVs)...
**************************************************
>>> Time elapsed : 0.0 sec
>>> Drop values list generated...
>>> Time elapsed : 0.6479 sec
>>> Drop values list generated...
>>> Time elapsed : 1.1786 sec
>>> Drop values list generated...
>>> Time elapsed : 1.6384 sec
>>> Drop values list generated...
>>> Time elapsed : 2.0312 sec
>>> Drop values list generated...
>>> Time elapsed : 2.3529 sec
>>> Drop values list generated...
>>> Time elapsed : 2.6173 sec
>>> Drop values list generated...
>>> Time elapsed : 2.8468 sec
>>> Drop values list generated...
>>> Time elapsed : 3.0406 sec
>>> Drop values list generated...
>>> Time elapsed : 3.1906 sec
>>> Drop values list generated...
>>> Time elapsed : 3.3161 sec
>>> Drop values list generated...
>>> Time elapsed : 3.424 sec
>>> Drop values list generated...
>>> Time elapsed : 3.5161 sec
>>> Drop values list generated...
>>> Time elapsed : 3.5985 sec
>>> Drop values list generated...
>>> Time elapsed : 3.6672 sec
>>> Drop values list generated...
>>> Time elapsed : 3.7272 sec
>>> Drop values list generated...
>>> Time elapsed : 3.7791 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8242 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8624 sec
>>> Drop values list generated...
>>> Time elapsed : 3.8944 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9216 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9446 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9644 sec
>>> Drop values list generated...
>>> Time elapsed : 3.982 sec
>>> Drop values list generated...
>>> Time elapsed : 3.9969 sec
```

```
>>> Time elapsed : 3.9909 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0103 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0225 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0331 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0429 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0514 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0579 sec
>>> Drop values list generated...
>>> Time elapsed : 4.064 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0702 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0762 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0821 sec
>>> Drop values list generated...
>>> Time elapsed : 4.088 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0939 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0997 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1054 sec
>>> Drop values list generated...
>>> Time elapsed : 4.111 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1165 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1218 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1268 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1313 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1354 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1387 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1417 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1444 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1469 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1493 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1515 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1534 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1551 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1567 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1581 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1594 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1605 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1613 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1621 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1628 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1634 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1639 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1644 sec
Breaking as no index was found at iter 63 ...
**************************************************
>>> Time elapsed : 0.0 sec
>>> Drop values list generated...
```

```
>>> Time elapsed : 0.578 sec
>>> Drop values list generated...
>>> Time elapsed : 1.2357 sec
>>> Drop values list generated...
>>> Time elapsed : 1.7639 sec
>>> Drop values list generated...
>>> Time elapsed : 2.1933 sec
>>> Drop values list generated...
>>> Time elapsed : 2.5782 sec
>>> Drop values list generated...
>>> Time elapsed : 2.9129 sec
>>> Drop values list generated...
>>> Time elapsed : 3.2004 sec
>>> Drop values list generated...
>>> Time elapsed : 3.454 sec
>>> Drop values list generated...
>>> Time elapsed : 3.688 sec
>>> Drop values list generated...
>>> Time elapsed : 3.871 sec
>>> Drop values list generated...
>>> Time elapsed : 4.0495 sec
>>> Drop values list generated...
>>> Time elapsed : 4.1842 sec
>>> Drop values list generated...
>>> Time elapsed : 4.3003 sec
>>> Drop values list generated...
>>> Time elapsed : 4.3997 sec
>>> Drop values list generated...
>>> Time elapsed : 4.4848 sec
>>> Drop values list generated...
>>> Time elapsed : 4.5616 sec
>>> Drop values list generated...
>>> Time elapsed : 4.6511 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7234 sec
>>> Drop values list generated...
>>> Time elapsed : 4.7737 sec
>>> Drop values list generated...
>>> Time elapsed : 4.8197 sec
>>> Drop values list generated...
>>> Time elapsed : 4.8606 sec
>>> Drop values list generated...
>>> Time elapsed : 4.8939 sec
>>> Drop values list generated...
>>> Time elapsed : 4.9213 sec
>>> Drop values list generated...
>>> Time elapsed : 4.9449 sec
>>> Drop values list generated...
>>> Time elapsed : 4.9655 sec
>>> Drop values list generated...
>>> Time elapsed : 4.9834 sec
>>> Drop values list generated...
>>> Time elapsed : 5.001 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0148 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0263 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0397 sec
>>> Drop values list generated...
>>> Time elapsed : 5.049 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0577 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0646 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0708 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0761 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0806 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0849 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0884 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0913 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0937 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0959 sec
```

```
>>> Drop values list generated...
>>> Time elapsed : 5.0978 sec
>>> Drop values list generated...
>>> Time elapsed : 5.0994 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1012 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1036 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1047 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1057 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1066 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1074 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1084 sec
>>> Drop values list generated...
>>> Time elapsed : 5.109 sec
>>> Drop values list generated...
>>> Time elapsed : 5.1095 sec
Breaking as no index was found at iter 53 ...
*** There are no missing elements in the target list...returning an empty index list.
*** Done POVs!
*** Attempting to compute variance inflation factors (VIFs)...

Legend   :
   C-Type: Column data type
   V-Type: Column value data type
   MLS   : Maximum length of [column values converted into] string data type
   NIL   : Null index list
   SCIL  : Search column index list (list of col-specific row indexes that match the search criterion)
   Mid   : Median
   POV   : Potential outlier value(s) for z-score thresh=1.5
   VIF   : Variance inflation factor(s)

Wall time elapsed: 128.4635 seconds
Wall time elapsed: 2.14 minutes
CPU time elapsed : 123.853 seconds
CPU time elapsed : 2.06 minutes
```

Out[17]:

	Feature	C-Type	V-Type	MLS	#NonNull	#Unique	#Null	SCIL	Min	Mid	Max	POV
0	date	object	[str]	1.00e+01	1.26e+06	1.15e+02	0.00e+00	N/A	2010/02/01	2015/05/01	2019/08/01	NaN
1	city	object	[str]	2.50e+01	1.26e+06	5.34e+03	0.00e+00	N/A	NaN	NaN	NaN	NaN
2	state	object	[str]	2.00e+00	1.26e+06	5.10e+01	0.00e+00	N/A	NaN	NaN	NaN	NaN
3	zip	int64	[int64]	5.00e+00	1.26e+06	1.09e+04	0.00e+00	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	6.12e+02	4.41e+04	9.99e+04	[612, 623, 692, 693, 725, 745, 778, 791, 802, ...
4	ppsf	float64	[float64]	7.00e+00	1.04e+06	6.43e+04	2.17e+05	N/A	1.05e+01	1.27e+02	2.80e+03	[10.46, 10.58, 11.08, 11.59, 11.73, 11.98, 11....

In [30]:

```
'''

To extract the row index or indexes with anomalous values from the mtab for the `zip`

column, you can use Pandas: mtab.loc[3, 'SCIL'] or mtab['SCIL'][3].

'''

zip_anomalous_idx = mtab['SCIL'][3]
print(f'anomalous value count: {len(zip_anomalous_idx)} | {zip_anomalous_idx[:20]}...')
```

```
anomalous value count: 115 | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]...
```

In [31]:
```
'''

With the dfsummarize method, you can also compute a frequency table (or ftab), comprising counts
and percentage share of total count of the top-n values by column of the given dataframe. The
default for n is 3. To show the ftab, pass True to the freq_table argument in the dfsummarize
method. In addition, you may include date_col_ls=[], num_col_ls=[], show_POV=False, and show_VIF=
False to reduce processing time. Specify ftab instead of mtab as the last line in this cell.

The raw frequency table (ftab) output is not sorted in the ascending order of column names. If you
care about the order in which the column names appear, the dfsummarize method allows you to sort
the ftab, specifically `Col` ascending and `Count` descending, by passing True to the argument
`sort_ftable` in dfsummarize. Add show_legend=False in dfsummarize if you don't want the mtab
legend to show when displaying the ftab.

The `city` column included in mtab, for example, is shown with top three values--New York, Los
Angeles, and Houston--under the `Value` column in the ftab. The 'CA' state count, which appears
as 116035 below, accounts for 9.24% of all values for the `state` column in the mtab.

'''

summary = time_func(dfz.dfsummarize, date_col_ls=[], num_col_ls=[], show_POV=False,
                    show_VIF=False, freq_table=True, sort_ftable=True, show_legend=False)
_, ftab = summary
ftab
```

```
*** Computing frequencies and proportions in the frequency table for the input dataframe...
    Dataframe shape (r, c): (1255915, 5)
    Features (cols) list  : ['date', 'city', 'state', 'zip', 'ppsf']
    Missing value count   : 216973

*** Searching data for missing values...
*** Eliminating missing values...
*** Row count has dropped from 1255915 to 1038942 due to null removal.
*** Post-row-removal null count across columns is 0.
*** Getting ready to compute the stats array...
*** Using ['zip', 'ppsf'] as the list of numerical columns for the stats array.

Wall time elapsed: 7.8455 seconds
Wall time elapsed: 0.13 minutes
CPU time elapsed : 7.4988 seconds
CPU time elapsed : 0.12 minutes
```

Out[31]:

	Col	Value	Count	Share
0	city	New York	13455	0.0107
1	city	Los Angeles	10120	0.0081
2	city	Houston	10005	0.0080
3	date	2010-02-01	10921	0.0087
4	date	2017-04-01	10921	0.0087
5	date	2017-02-01	10921	0.0087
6	ppsf	125.0	488	0.0005
7	ppsf	100.0	478	0.0005
8	ppsf	104.17	373	0.0004